Adaptive Control of UAVs in Close-Coupled Formation Flight

Dr. Corey J. Schumacher
AFRL/VACA
2210 8th Street, Bldg 146, Room 305
Wright-Patterson AFB OH 45433-7531
corey.schumacher@wpafb.af.mil

Rajeeva Kumar
Department of Aerospace Engineering
University of Michigan
Ann Arbor, MI 48105
rajeevak@umich.edu

ABSTRACT
This paper studies the control of multiple UAV's flying in a close-coupled formation for the purposes of drag reduction. A controller design methodology for use in the trail vehicle in a two-UAV formation is presented. The LQR outer-loop tracks relative position commands and generates body-axis rate commands for the inner loop. The adaptive dynamic inversion inner-loop tracks these input commands using only minimal knowledge of the aircraft dynamics. The controller is tested in a two-vehicle formation flight simulation. Excellent command tracking and performance are achieved without use of specific knowledge of the formation flight effects. Simulation results demonstrate that the proposed controller design enables the trail UAV to maneuver in the lead UAV's wake, and to hold a desired position in the vortex.

1.0 Introduction

It is known that aircraft with large aspect ratio wings have better overall aerodynamic efficiency because of reduction in induced drag for a given lift. However, large aspect ratio implies large wingspan for a given area. This means for lightweight design the resulting structure will be unreasonably flexible and fragile. A similar improvement in overall efficiency can be achieved by flying multiple aircraft in close formation. In an idealized case of n identical aircraft, each with aspect ratio a, flying in wing-tip to wing-tip formation the effect would be that of a single craft with $n.a$ aspect ratio. This kind of drag reduction in close formation is due to favorable wake-vortex encounters. Wind tunnel tests have shown that this drag benefit is real and analytical studies predict that the benefit increases as additional aircraft are added to the formation

A number of recent studies have examined the problem of close-coupled formation flight for multiple aircraft. The formation control of linear aircraft models has been considered [1-3]. Formation control of two aircraft using PID feedback has been presented in [1], where it is assumed that Mach-, altitude-, and heading-hold autopilots are available, and only the outer-loop problem is examined. A linear decentralized controller is presented in [2]. In [3], an optimal peak-seeking controller is presented, which attempts to find the optimal location for minimum drag during aircraft flight. These studies have all been limited to linear aircraft models.

The modeling of vortex-effects on the aircraft in close-coupled formation flight is also a subject of considerable recent interest, and is very important for effective control system design. These aerodynamic coupling effects are highly nonlinear and difficult to model accurately. The results of wind tunnel experiments to measure forces and moments on a trail aircraft in a vortex are presented in [4].

This paper examines the controller design problem for an unmanned aerial vehicle (UAV) flying in the wake of another, identical UAV for the purposes of drag reduction. This is a particularly challenging control problem for a number of reasons. The vortex generated by the lead vehicle induces large, nonlinear forces and moments on the trail (or chase) vehicle. The induced rolling moment, in particular, is quite large and highly nonlinear, and varies substantially in both magnitude and sign as the trail aircraft changes position in the vortex. Additionally, the formation flight effects are very difficult to measure and to model. Detailed accurate knowledge of the formation flight effects is generally not available during flight.

The controller developed here is a two-loop controller. The outer-loop is an LQR controller that tracks incoming lateral and vertical position commands and generates power lever angle and body axis rates commands. The inner-loop controller is a dynamic inversion controller with an adaptive neural network that takes body axis p, q, r, commands as inputs and generates control surface deflection commands. The adaptive dynamic inversion method used in the inner loop is developed in detailed in [5-8].

This paper presents an effective control design method for close-coupled formation flight. Simulation results show that the controller achieves good tracking of position commands. The adaptive dynamic inversion inner-loop tracks the p-, q-, r-commands very well, even though it uses only very limited information about the system dynamics.

The organization of this paper is as follows. Section 2 provides details of the formation flight model used in the simulation. In Section 3, the formation flight control system is discussed, and simulation results are presented in Section 4.

2.0 Formation Flight Modeling

A six-degree of freedom nonlinear simulation model has been developed in Matlab/Simulink. The nonlinear equations of motion for an aircraft were used, but the aerodynamic coefficients were held constant. This restriction does not greatly reduce the validity of the simulation results because only relatively small maneuvers (with approximately constant aerodynamic coefficients) are performed in the simulation results presented. A table look-up for the formation flight data was also included, containing the changes in all force and moment coefficients as functions of relative X, Y, and Z separation between the lead and chase vehicles. This model was generated for 1-g straight and level flight trimmed at Mach 0.8 and altitude 45000 ft. The relative position of the lead vehicle with respect to the chase vehicles is expressed in an inertial, right-handed reference frame with the X-axis out the lead aircraft's nose, the Y-axis out the right wing, and the Z-axis down.

The formation flight data were given in the following form:

$$C_{L,i} = C_{L,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{L,i}}{C_{L,j}}\right)C_{L,j}$$

$$C_{D,i} = C_{D,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{D,i}}{C_{L,i}C_{L,j}}\right)C_{L,i}C_{L,j} \qquad (1)\text{-}(6)$$

$$.C_{m,i} = C_{m,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{m,i}}{C_{L,j}}\right)C_{L,j}$$

$$C_{l,i} = C_{l,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{l,i}}{C_{L,j}}\right)C_{L,j}$$

$$C_{n,i} = C_{n,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{n,i}}{C_{L,i}C_{L,j}}\right)C_{L,i}C_{L,j}\ ,$$

$$C_{Y,i} = C_{Y,i}(s) + \sum_{j,i\neq j}\left(\frac{\Delta C_{Y,i}}{C_{L,i}C_{L,j}}\right)C_{L,i}C_{L,j}$$

where the terms in the parenthesis are provided in table look-up as a function of relative distances. It should be noted that the coupling effect is given as an add-on to six unperturbed aerodynamic forces and moments.

This model provides a great deal of information about the vortex effects during close-coupled formation flight, but extensive modeling work is still required. Time delay effects and the effects of aircraft rolling motion, as well as other, smaller, effects still need to be included.

The primary effects of the aerodynamic coupling are on drag, lift, and rolling moment experienced by the trail aircraft. The lead aircraft does experience some aerodynamic coupling, but it is relatively minor. The formation flight effects are highly nonlinear, and are extremely difficult to model completely.

3.0 Formation Flight Control

The formation flight controller presented in this paper is a controller for the chase vehicle only. It is designed to track relative y- and z- position commands while holding the vehicle's speed constant. There is an implicit assumption that the position of the lead vehicle is known so that appropriate y- and z-commands can be given. It is composed of two major parts: an outer-loop LQR controller and an inner-loop adaptive dynamic inversion controller. The outer loop takes relative y and z position commands and generates Power Lever Angle (PLA) commands and body axis rotation rate commands (p,q,r commands). The p,q,r commands are input to the inner-loop dynamic inversion control, which generates control surface deflections to achieve the desired body axis rotation rates.

3.1 Outer Loop – Optimal LQR design

The outer loop controller is an LQR design, taking lateral and vertical position commands as inputs and using PLA and body rates p,q,r as control inputs to achieve the desired commands. The controller also must maintain stability and keep the aircraft's flight speed constant. The controller is designed using a linear model of the trail UAV trimmed in the lead UAV's wake. This linear model has states x = $(\alpha, \beta, V, \phi, \theta, \psi, y, z)^T$, where y and z are the lateral and vertical position of the vehicle in an inertial frame. The control inputs to the linear model are the body axis rates p,q,r and power lever angle, PLA. The state vector is augmented with integral error states for V, y, and z, so that V-error can be minimized and y and z commands can be tracked. The system dynamics for the LQR control design problem are then of the form:

$$\dot{x} = Ax + Bu + B_1 w \qquad (7)$$

with

$x = (\alpha, \beta, V, \phi, \theta, \psi, y, z, \int V_c - Vdt, \int y_c - ydt, \int z_c - zdt)^T$,
$u=(p,q,r,PLA)^T$, and $w=(y_c, z_c)^T$.

The performance index to be minimized is defined as

$$J = \frac{1}{2}\int (x^T Qx + z^T Rz)dt \qquad (8)$$

Where Q and R are diagonal weighting matrices as described below.

$$Q = diag(q_i), i = 1,...,11 \qquad (9)$$
$$R = diag(r_i), i = 1,...,4$$

Thus, the control, u, for the outer loop turns out to be:

$$u = -R^{-1}B^T Px \qquad (10)$$

Where P is the positive semi-definite solution of the Riccati equation.

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \qquad (11)$$

The weights used for Q and R were:
$q_i = (100,500,0.1,1,0.1,0.1,5,0.1,10,5,0.5)^T$,
$r_i = (1000,500,1000,100)^T$.
Certain states, particularly angle of attack, sideslip angle, and the "controls" p, q, and r were weighted very heavily,

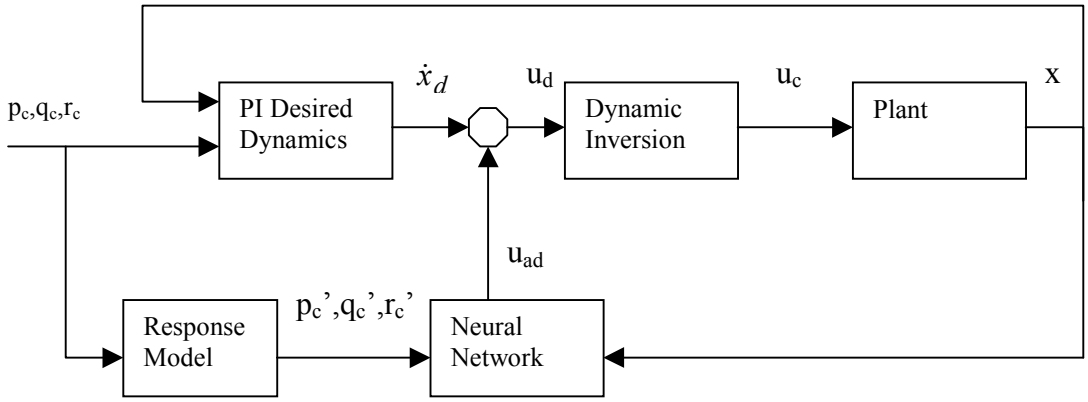

Figure 1: Inner-loop Adaptive Dynamic Inversion Controller

as large excursions in these states quickly cause instability in the presence of the formation flight effects The outer-loop control is then formed by $u=-Kx$, where $K=R^{-1}B^TP$ is the 4x11 gain matrix calculated from (11). The first three elements of u are passed to the inner loop controller, which calculated the control surface deflections. The fourth element is used directly as the "throttle" command. In the simulation results in this paper we are only trying to maintain constant flight speed, so V_c is zero.

3.2 Inner Loop – Adaptive Dynamic Inversion

The inner-loop controller uses a nominal, implicit-command following dynamic inversion controller augmented by an adaptive neural network to compensate for inversion errors. This method was first developed in [5-7] for explicit command-following dynamic inversion and was extended in [8] to an implicit command-following controller. The inner loop controller takes body rate ***p,q,r*** commands as inputs and generates control surface commands to ensure tracking of the input rate commands. In dynamic inversion control, we seek to linearize a nonlinear system [9]. The aircraft's dynamics can be put in the form:

$$\dot{\mathbf{x}} = \mathrm{f}(\mathbf{x}) + \mathrm{g}(\mathbf{x})\mathbf{u}, \tag{12}$$

where **x** is a vector the controlled states and **u** is a vector of controls and f(**x**) and/or g(**x**) is nonlinear. The number of controlled states and controls need to be the same (a square system). The control can be calculated according to

$$\mathbf{u}_c = \mathrm{g}^{-1}(\mathbf{x})(\mathbf{u}_d - \mathrm{f}(\mathbf{x})), \tag{13}$$

where $\mathbf{u}_d$ is the desired response of $\dot{\mathbf{x}}$. Substituting (2) into (1), for **u**, results in

$$\dot{\mathbf{x}}_d = \mathbf{u}_d \tag{14}$$

and any nonlinearities in f(**x**) and g(**x**) are cancelled. Figure **2** shows a diagram of the inner-loop adaptive dynamic inversion controller. In the figure, the body rotation-rate commands from the outer loop are the inputs. These inputs are used to calculate the desired dynamics, and are also passed into the first order command filter for use in the neural network weight adaptation law. In this application, we use an implicit command-following controller with proportional-integral desired dynamics, so the desired dynamics $\mathbf{u}_d$ become:

$$\mathbf{u}_d = \tfrac{1}{2}K_b\mathbf{x}_c - K_b\mathbf{x} + \tfrac{1}{4}K_b^2\int(\mathbf{x}_c - \mathbf{x})dt\,. \tag{15}$$

The desired dynamics are identical in all three channels, roll, pitch, and yaw.

If the functions f(x) and g(x) are not known exactly, then the inversion will not be performed exactly, resulting in inversion error. In this case, the resulting dynamics are

$$\dot{x} = \tfrac{1}{2}K_b\mathbf{x}_c - K_b\mathbf{x} + \tfrac{1}{4}K_b^2\int(\mathbf{x}_c - \mathbf{x})dt + \Delta\,, \tag{16}$$

where Δ is the inversion error.

The nominal dynamic inversion is performed using only a linear model of the aircraft dynamics, calculated at the initial trim condition with the trail vehicles flying in the lead vehicles wake. The nonlinear vehicle dynamics are not included in the inversion model, and no knowledge of the formation flight effects are used in the inversion, other than that incorporated in the linear model calculated at the initial trim flight condition. This simple dynamic inversion is easy to implement, but results in large inversion errors, for which the neural network must compensate.

This adaptive inner-loop controller requires only minimal knowledge of the vehicle dynamics. As will be seen in the simulation results, this adaptive inner loop tracks the desired p,q,r commands very well, even though the inversion is calculated with very little information about the formation flight and nonlinear coupling effects. This adaptive control strategy was chosen for the inner loop specifically because of its ability to achieve tracking with highly uncertain dynamics.

3.3 Adaptive Neural Network

In this study we use a linear-in-parameters neural network. An identical network structure was used in all three channels, with only a slight variation in the inputs. The inputs to each network were the vehicle's body axis rates p, q, and r, the squashed control for that control channel, $\sigma(u_i)$,

where $\sigma(u_1) = \frac{2}{1+e^{-0.1u_1}} - 1$. The purpose of the squashing function is to guarantee that the control-related input to the neural network has a magnitude less than one, guaranteeing a solution to the algebraic loop around the network. In this application, all of the network inputs were excursions from trim values, not the full values of the states/controls. The basis functions in each network were linear in the normalized inputs, without cross terms. Thus, the basis functions used were *1, p, q, r, Δy, Δz, σ(u_i)*, resulting in seven weights and seven basis functions in each of the three networks.

An important part of any neural network is the rule governing its adaptation. In the method used here, the weight adaptation law is derived via Lyapunov stability analysis, and is dependent on the desired dynamics in the dynamic inversion. In [8], it is shown that the appropriate weight-update law for use with an implicit command-following dynamic inversion controller with desired dynamics given by (5)

$$\dot{w}_{pi} = \gamma\left(\left(x_c' - x\right) b_i + \eta \left| x_c' - x \right| w_i \right) \qquad (17)$$

where w_i is the ith weight of the linear-in-parameters neural network, γ is learning rate, b_i is the ith basis function, and η is an *e-modification* factor to unsure that **w** is bounded. In this application, the gains were chosen as γ=2500 and η = 0.05. In (9), x_c' is the output of a first order command filter with transfer function $\frac{x_c'}{x_c} = 0.5k_b \frac{1}{s+0.5k_b}$. This response filter is identical to the desired dynamics (5) except for a stable pole-zero cancellation, and is necessary for the weight update law to be correct. Further information on the use of this filter can be found in [8]. This neural network results in an adaptive controller, with no long-term memory. No off-line learning is required with this adaptive network, because it has no long-term memory. With the weight update law given by (9), closed-loop stability and boundedness of the control signals are guaranteed.

The output of the network is formed by $u_{ad} = b^T w$, where b is the vector of basis functions and w is the vector of weights. Ideally, the output of the network will exactly match the inversion error, resulting in exact dynamic inversion. This output is subtracted from u_d in (15) to compensate for the inversion error**.** Then (16) becomes

$$\dot{x} = \frac{1}{2}K_b \mathbf{x}_c - K_b \mathbf{x} + \frac{1}{4}K_b^2 \int (\mathbf{x}_c - \mathbf{x}) dt + \Delta - u_{ad} \quad .(18)$$

If $u_{ad} = \Delta$, then the network exactly matches the inversion error and the desired dynamics are achieved exactly.

4.0 Simulation Results

The multiloop controller detailed in the previous sections was tested in the formation flight simulation with two UAVs flying in formation at mach 0.8 and an altitude of 45,000 ft. The simulation was initialized with both UAV's in a trim condition, with the trail UAV 80 feet behind, 3 feet above, and 33.75 feet to the side of the lead vehicle. The x-position was chosen so that there would be adequate longitudinal separation between the two vehicles for safety purposes. The y-position was chosen to be 90% of the aircraft wingspan, which is approximately the optimal position for drag reduction. A small positive z-separation was chosen because the relative z-position of the trial vehicle is stable if it is slightly above the lead vehicle. In this case, increasing altitude of the trail vehicle results in less induced lift, and a decrease in altitude gives increased induced lift.

Throughout the simulation the lead aircraft is simply commanded to continue in straight and level flight, so no results are shown for the lead vehicle. There are only very small variations in the lead aircraft's state, due to the minor effects of formation flight on the lead vehicle. One second after the simulation begins, the chase vehicle is commanded to move 5 feet farther away from the lead vehicle in both lateral and vertical position, and at t=10sec it is commanded back to its original position in approximately the optimal position in the lead aircraft's vortex.

The inertial Y and Z positions of the chase aircraft are shown in Figure 2. For both Y- and Z-commands, the vehicle achieves the desired position in approximately 4 seconds, with small overshoot and good settling times. The controller does an excellent job of tracking the input commands, even in the presence of the large nonlinear effects of formation flight.

The inner-loop responses are shown in Figure 3. In these plots, both the filtered inputs to the inner loop (e.g. p_c'), and the state responses are shown, but in all three plots only a single line can be seen. The adaptive inner-loop dynamic inversion controller achieves almost exactly the nominal desired dynamics, even though only a single linear model is used to calculate the inversion. The adaptive neural network does an excellent job of compensating for the inversion errors caused by formation flight and other nonlinearities in the aircraft dynamics.

This is a particularly valuable result, because the moment effects due to formation flight, particularly the rolling moment, are large and highly nonlinear. It is very difficult to design a linear or non-adaptive controller for the inner loop, due to the severe changes in the dynamics as the trail aircraft changes y-position relative to the lead aircraft.. During formation flight, the induced forces and moments will always be highly uncertain, making an adaptive controller that needs little direct knowledge of these forces and moments critical. With only minimal knowledge of the aircraft dynamics (a single linear model), the inner loop adaptive dynamic inversion achieves the desired dynamics almost exactly. Without the adaptive neural network, the linear dynamic inversion controller cannot control the

aircraft. Very large tracking errors appear in the inner-loop and the aircraft quickly goes unstable.

The other states of the trail UAV are not shown due to space limitations. There are only small excursions in angle-of-attack, sideslip angle, flight speed, and the euler angles. The trail UAV's response is well-behaved while changing position in the vortex of the lead vehicle. The relative x-separation between the two vehicles it remains within 7 inches of the desired initial separation of 80 feet throughout the simulation.

5.0 Conclusions and Future Work

This paper presented a formation flight controls system for UAVs that consisted of an inner-loop adaptive dynamic inversion controller and an outer-loop LQR controller. This controller was used in the position control of one UAV flying in the wake of a second UAV. The adaptive inner-loop dynamic inversion controller achieved excellent tracking of p,q,r commands in the presence of large nonlinear formation flight effects, even though only a single linear model was used to perform the nominal inversion. The overall controller performed well, enabling the trailing UAV to achieve and maintain desired positions in the wake of the lead UAV.

There are many promising directions for future work in this area. A full formation controller still needs to be developed, directly controlling the longitudinal (x-direction) separation between the two aircraft, instead of simply maintaining constant flight speeds. Also, output feedback with realistic measurement errors need to be examined. Another important extension of this work will be to apply it to formations of three or more aircraft, so that string stability issues can be examined. More detailed models of the formation flight vortex effects are also needed. Both the modeling and control work discussed in this paper also have application to aerial refueling problems for unmanned vehicles, and this area should also be pursued.

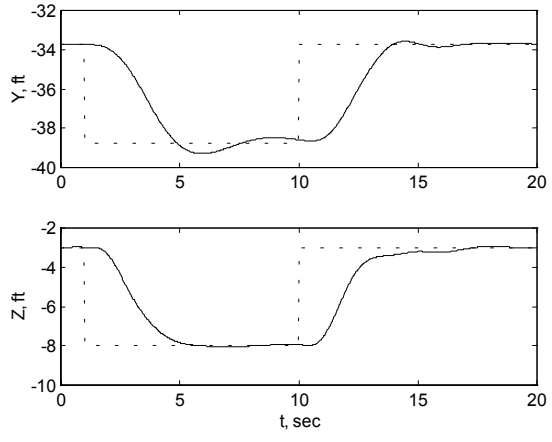

Figure 2: Lateral and Vertical Response of Trail UAV to Relative Displacement Commands

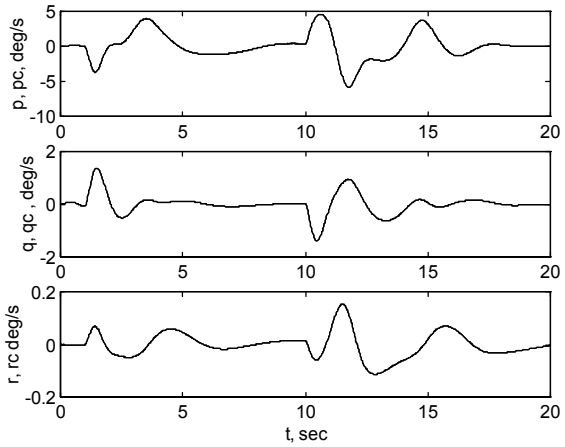

Figure 3: Inner-loop Command Response during Maneuver

References

1. Proud, A., Pachter, M., and D'Azzo, J. J., "Close Formation Control," Proceedings of the AIAA Guidance, Navigation, and Control Conference, AIAA – 99 – 4207, pp. 1231-1246, Portland, OR, August 1999.

2. Wolfe, J.D., Chichka, D.F., and Speyer, J.L., "Decentralized Controllers for Unmanned Aerial Vehicle Formation Flight," in Proceedings of the AIAA Guidance, Navigation, and Control Conference, San Diego, CA, July 1996, 29-31.

3. Chichka, D.F., Park, C.G., and Speyer, J.L., "Peak-Seeking Control with Application to Formation Flight," in Proceedings of IEE Conference on Decision and Control, December 1999.

4. Myatt, J.H., and Blake, W.B., "Aerodynamic Database issues for Modeling Close Formation Flight, in Proceedings of the AIAA Modeling and Simulation Technologies Conference, Portland, OR, August 1999.

5. Kim, B.S. and A.J. Calise, Nonlinear flight control using neural networks. Journal of Guidance, Control, and Dynamics, 1997. 20(1): p. 26-33.

6. Leinter, J., Calise, A., and Prasad, J.V.R., Analysis of Adaptive Neural Networks for Helicopter Flight Control, Journal of Guidance, Control, and Dynamics, 1997, 20(5): p. 972-979.

7. Calise, A.J., Rysdyk, R.T., Nonlinear Adaptive Flight Control Using Neural Networks, IEEE Control Systems Magazine, Dec 1998, V 18 n 6.

8. Schumacher, C. and Johnson, J. D., "PI Control of a Tailless Fighter Aircraft with Dynamic Inversion and Neural Networks," in Proceedings of the 1999 American Control Conference, pp 4173-4177.

9. Reiner, J., G.J. Balas, and W.L. Garrard, Robust dynamic inversion for control of highly maneuverable aircraft. Journal of Guidance, Control, and Dynamics, 1995. 18(1): p. 18-24.